Lane Keeping with Misaligned Rear Wheels

Nolan Wagener

Motivation

Lane keeping analysis for front-wheel steering cars is usually done assuming that the rear wheel is not turned. Of course, this is an idealization, and in the real world the rear wheels are most likely not parallel to the car body. For most cars just out of a factory, the deviation is probably negligible. However, there may be some cars that slip through quality control with significant defects, or a car may have been damaged in a car accident or through faulty auto repair. The effects may either be misaligned rear wheels or some situation such as greater weight on one side of the car that would effectively make the rear wheels seem misaligned.

Thus far, open-loop analysis has been done and controller schemes have been designed assuming that the rear wheel has zero steering angle. When that is no longer the case, though, how much does that affect our models' and controllers' performances which were made with the faulty assumption? If significant, can we revise our controller design so that even with misaligned rear wheels, the vehicle can still perform lane keeping properly?

Modeling

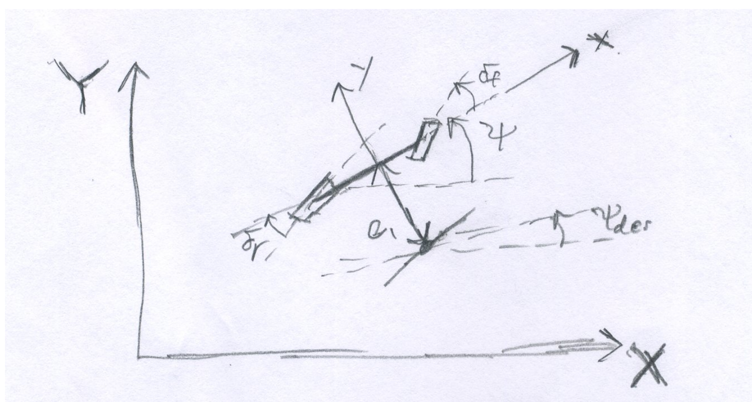

Figure 1: Two Degree of Freedom Bicycle Model

We will use the two degree of freedom bicycle model for the dynamics of a vehicle, as shown in Figure 1. The standard equations of motion for the vehicle's lateral dynamics are:

$$m\ddot{y} + \frac{2C_{\alpha_f} + 2C_{\alpha_r}}{V_x}\dot{y} + \left(mV_x + \frac{2\ell_f C_{\alpha_f} - 2\ell_r C_{\alpha_r}}{V_x}\right)\dot{\psi} = 2C_{\alpha_f}\delta_f + 2C_{\alpha_r}\delta_r \qquad (1)$$

$$I_{zz}\ddot{\psi} + \frac{2\ell_f C_{\alpha_f} - 2\ell_r C_{\alpha_r}}{V_x}\dot{y} + \frac{2\ell_f^2 C_{\alpha_f} + 2\ell_r^2 C_{\alpha_r}}{V_x}\dot{\psi} = 2\ell_f C_{\alpha_f}\delta_f - 2\ell_r C_{\alpha_r}\delta_r \qquad (2)$$

The meanings of the variables in the equation can be found in Appendix B.

Canonical Examples

Before diving into the theory, let's use a few examples to visualize the effects of a nonzero rear steering angle. Let's say we have a car with the following properties: $m = 1573$ kg, $I_{zz} = 2873$ kg·m^2, $\ell_f = 1.1$ m, $\ell_r = 1.58$ m, and $C_{\alpha_f} = C_{\alpha_r} = 80000$ N/rad. The rear wheels are misaligned by some amount, realistically between $-2°$ and $2°$. We will assume that the road-tire interface can provide sufficient friction for our simulations. As well, we'll assume that there is no roll effects on our vehicle.

We are given two scenarios: 1) driving in a straight line, and 2) driving in a circle of radius 250 meters, both at longitudinal speed 20 m/s (45 mph).

We will simulate these scenarios on MATLAB, analyzing the lateral and yaw errors, along with the global position (X, Y) of the car. The global position can be found by integrating the global velocities ($\dot{X}$, $\dot{Y}$) over time. The global velocities are given in Rajamani's text:

$$\begin{aligned} \dot{X} &= V_x \cos(\psi + \beta) && (3) \\ \dot{Y} &= V_x \sin(\psi + \beta) && (4) \end{aligned}$$

where $\beta = \arctan\left(\dfrac{\ell_f \tan\delta_r + \ell_r \tan\delta_f}{L}\right)$. Equations 1 through 4 along with initial conditions, system properties, and inputs will give us the trajectory of a vehicle.

To best analyze and simulate the system, equations 1 and 2 are converted into error coordinates and into state space. Our state will be a vector of errors, $\mathbf{x} = [e_1 \quad \dot{e}_1 \quad e_2 \quad \dot{e}_2]^T$, where e_1 is the distance between the car's center of gravity and the lane's center line, and e_2 is the difference between the yaw and desired yaw. Taking an equation from Rajamani's text and including the δ_r term, we get

$$\dot{\mathbf{x}} = A\mathbf{x} + B_1\delta_f + B_2\delta_r + B_3\dot{\psi}_{des} \tag{5}$$

with the following defined matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{2(C_{\alpha_f} + C_{\alpha_r})}{mV_x} & \dfrac{2(C_{\alpha_f} + C_{\alpha_r})}{m} & \dfrac{2(-C_{\alpha_f}\ell_f + C_{\alpha_r}\ell_r)}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\dfrac{2(C_{\alpha_f}\ell_f - C_{\alpha_r}\ell_r)}{I_{zz}V_x} & \dfrac{2(C_{\alpha_f}\ell_f - C_{\alpha_r}\ell_r)}{I_{zz}} & -\dfrac{2(C_{\alpha_f}\ell_f^2 + C_{\alpha_r}\ell_r^2)}{I_{zz}V_x} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ \dfrac{2C_{\alpha_f}}{m} \\ 0 \\ \dfrac{2C_{\alpha_f}\ell_f}{I_{zz}} \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ \dfrac{2C_{\alpha_r}}{m} \\ 0 \\ -\dfrac{2C_{\alpha_r}\ell_r}{I_{zz}} \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 \\ -\dfrac{2(C_{\alpha_f}\ell_f - C_{\alpha_r}\ell_r + \frac{1}{2}mV_x^2)}{mV_x} \\ 0 \\ -\dfrac{2(C_{\alpha_f}\ell_f^2 + C_{\alpha_r}\ell_r^2)}{I_{zz}V_x} \end{bmatrix}$$

We note that $\dot{\psi}_{des} = \dfrac{V_x}{R}$ and δ_r will be some constant value.

We will now analyze the simulation when the desired path is a straight line.

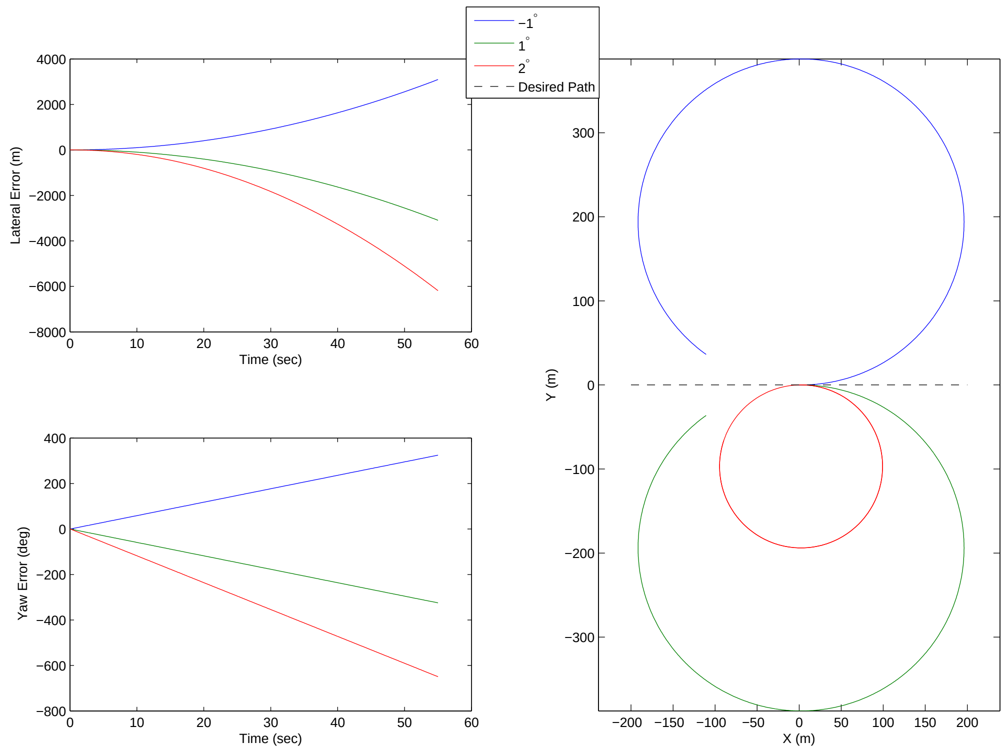

Figure 2: Open Loop Result with Straight Line Path

Figure 2 shows the result of simulating with a straight line input with different rear steering angles. Each resulting trajectory is a circle, with a steering angle resulting in a smaller circle. Thus, there is an inverse relation between the radius of the path and the rear steering angle. Even a steering angle as small as 1 degree obviously makes an astounding difference in the deviation from the path.

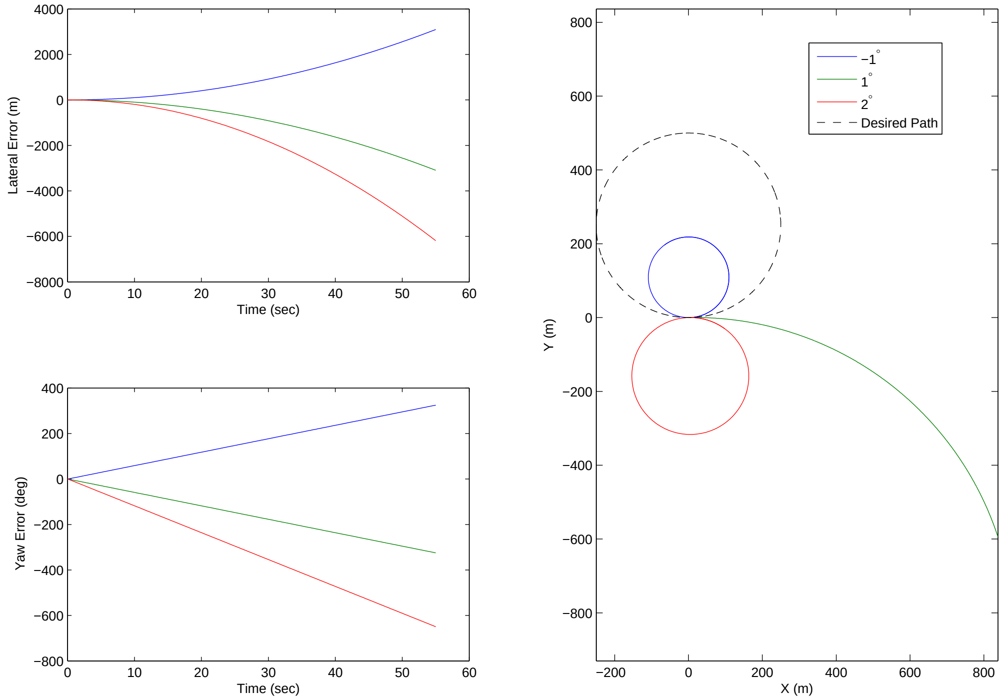

Figure 3: Open Loop Result with Circular Path

Figure 3 shows the result of simulating with a circular road input with different rear steering angles. Once again, small values of δ_r cause large deviations in the path, with the car going in the wrong direction for large enough values of steering angle.

Effect of Misaligned Rear Wheels

As shown by the examples, nonzero steering angles significantly deter lane keeping in the open loop. We will know analyze our model to pin down how significant a role the rear steering angle plays in general.

Say our desired trajectory is a circle of radius R_0 and we assume that $\delta_r = 0$. From Appendix A, equation 13, we find that

$$\delta_f = \frac{L + K_{US}\frac{V_x^2}{g}}{R_0}$$

Of course, in our scenario the car will not trace out a circle of radius R_0, but will instead trace a circle of radius R due to the nonzero rear steering angle (explanation in Appendix A). Let $L_e = L + K_{US}\dfrac{V_x^2}{g}$ represent the effective length of the car as function of the car's length and speed.

Thus, $\delta_f = \frac{L_e}{R_0}$ and, from Equation 13,

$$\begin{aligned} R &= \frac{L_e}{\delta_f - \delta_r} \\ &= \frac{L_e}{\frac{L_e}{R_0} - \delta_r} \end{aligned}$$

After a little simplification we get

$$R = \frac{L_e}{L_e - R_0\delta_r} R_0 \tag{6}$$

Looking at equation 6, what effects does δ_r have?

- A positive δ_r increases the radius, and a negative δ_r decreases the radius.
- A larger nominal radius R_0 will cause R to deviate more from R_0. A smaller R_0 will cause R to be closer to R_0.
- A faster speed V_x will cause R to be closer to R_0. However, if K_{US} is negative, there is an upper limit on the speed, specifically at $V_{cr} = \sqrt{\frac{gL}{-K_{US}}}$, before instability occurs. As well, increasing V_x while keeping the same R_0 will cause δ_f to increase for positive K_{US} or decrease for negative K_{US}. Thus, any open-loop improvements in R will cause changes in other parameters such as δ_f.

We will illustrate what increasing V_x (while correctly adjusting δ_f) will do, as it affects more than R. Shown in Figure 4 are the simulation results of driving in a circle as the longitudinal speed increases. As expected, the radius of the path R does approach the radius of the road R_0. However, a tilt develops between the car's path and the desired path. The tilt occurs as a result of the transient response, wherein the car begins to veer in a different direction. Since there is no controller pushing the car back onto the road, the car's trajectory will converge to a circle, as expected, but with an offset from the desired one.

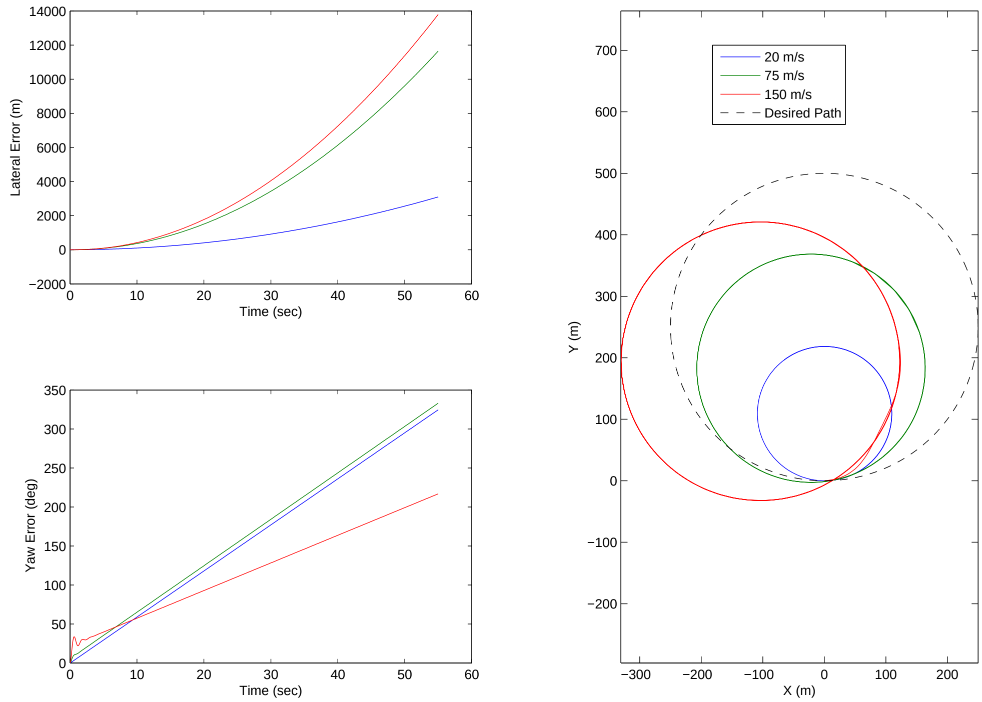

Figure 4: Effects of Increasing Speed

It is also interesting to note an equivalent effect a nonzero δ_r has. Letting $\delta = \delta_f - \delta_r$ and rearranging equation 13 gives

$$\delta = \frac{L}{R} + K_{US}\frac{V_x^2}{gR} \tag{7}$$

This is the same equation as the one that finds the desired front steering angle for a trajectory of radius R assuming $\delta_r = 0$. Thus, having a steering configuration δ_f and δ_r is equivalent in terms of the path's radius to the front wheel having steering angle $\delta_f - \delta_r$ and the rear wheel having no steering angle. This is illustrated in Figure 5. So if you want to drive straight (as is most often the case), have $\delta_f = \delta_r$, i.e. drive with the front wheel turned the same amount as the rear wheel. This does not mean though that both vehicles illustrated are the same. Driving with both wheels at $0°$ will have a different transient response than having both wheels at $1°$.

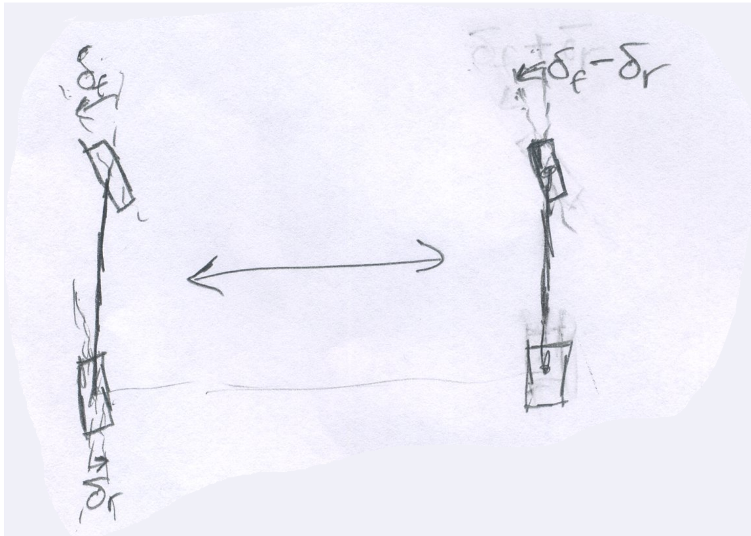

Figure 5: Steady State Equivalence of Steering Angles

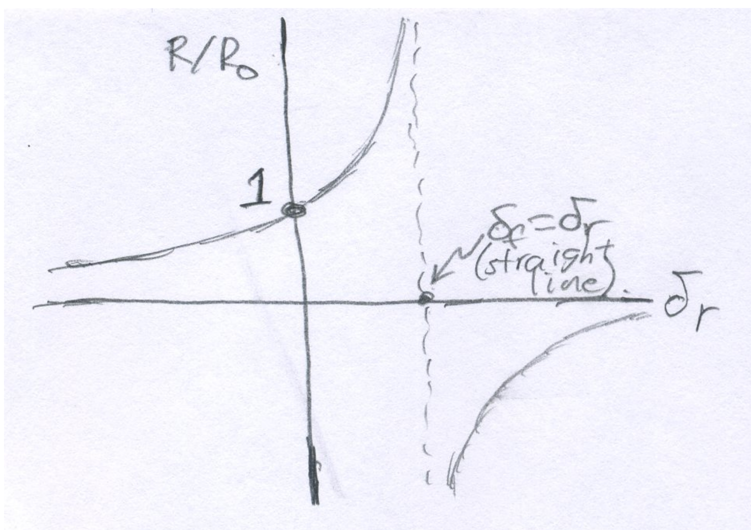

Figure 6: R/R_0 versus δ_r

A graphical representation of R/R_0 is given in Figure 6. When $\delta_f = \delta_r$, R becomes infinite and the car drives on a straight line. Once R/R_0 becomes negative, the car will drive in an opposite direction (such as going from driving counterclockwise to driving clockwise) but will still trace a circle. Although the plot asymptotes to 0 on both sides, in reality this would not happen. One main assumption in deriving the fundamental equations 1 and 2 are that δ_f and δ_r are small. In fact, a more realistic plot of R/R_0 vs. δ_r would be periodic with π radians. This is because a tire will look the same after being rotated π radians or 180°. We won't have to worry about such effects though as we will limit $|\delta_r|$ to be under 2°, so that the small angle approximation is still valid.

In general, if you have knowledge of the value of δ_r and find that to drive with a trajectory of radius R_0 (calculated assuming $\delta_r = 0\,!$), just add δ_r to δ_f, and that is your proper front steering angle. Of course, we won't in general know that such a δ_r exists, let alone its value, so such an open-loop solution doesn't bear much practical value. If one knows their rear wheels are misaligned, better to have the car fixed than to deal with compensating for the misalignment while driving.

As well, even just adding the rear steering angle to the front steering angle only works in steady state. When driving on a straight road, having $\delta_r = \delta_f$ will allow the car to drive straight, but the car will not keep on the road. Figure 7 shows such a simulation result. Each of the yaw errors converge this time around (implying the car's yaw converges), but since they are nonzero the car slowly diverges from the road. Without more correction, the car would realistically drive off the road in a matter of seconds.

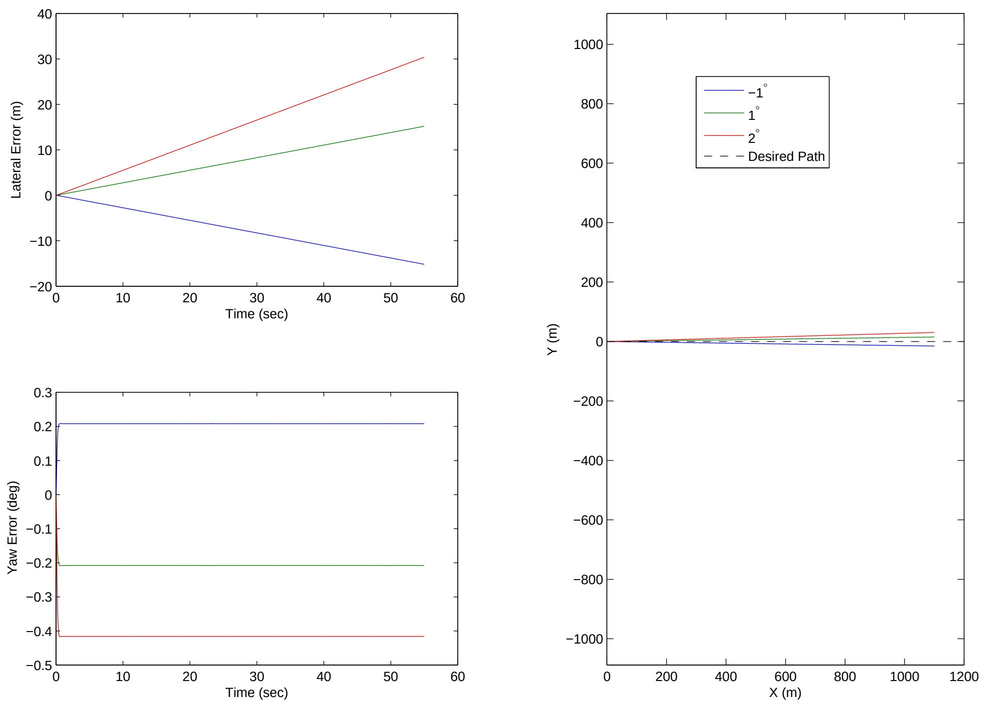

Figure 7: Simple Compensation Result with Straight Path

Figure 8 shows simulation results when driving on a circular road, and the results are almost identical to that of Figure 7. The car creeps off the circular road in the same manner as in the straight road and will likely drive off the road in a matter of seconds.

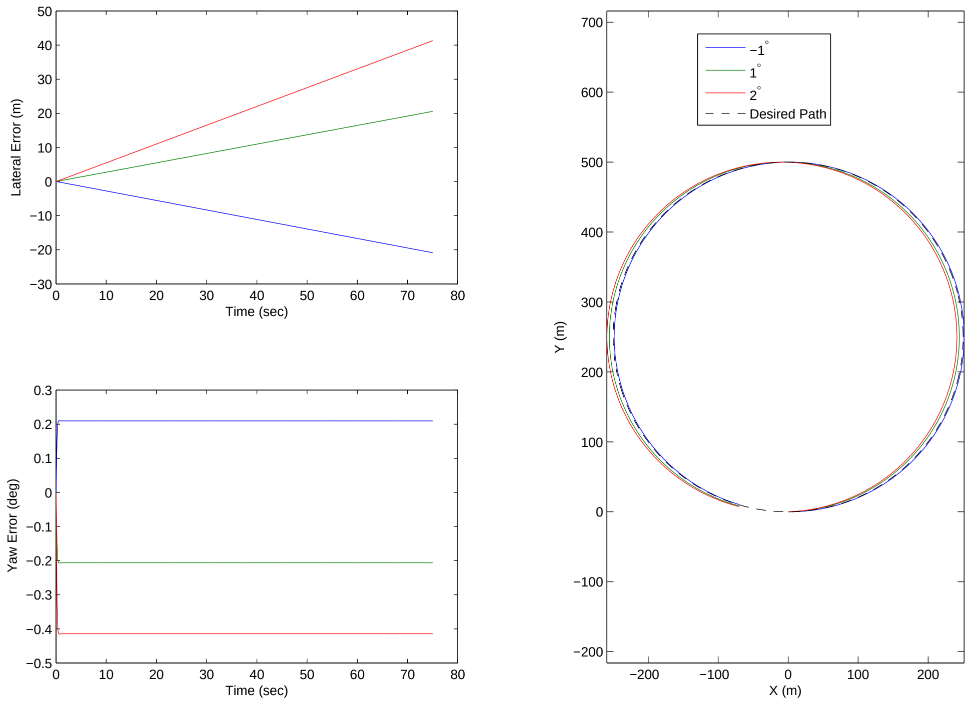

Figure 8: Simple Compensation Result with Circular Path

Thus, even a simple compensation scheme as mentioned won't work in the long run due to the car's transient response. Something more sophisticated is needed to keep a car within its lane.

Closed-Loop Solution: State Feedback Controller

So can we design a controller that can keep a car on a desired path without knowledge of the disturbance δ_r? Let us attempt to control the system using state feedback along with a feedforward term.

We refer to our state space model in Equation 5. We then define our input δ_f to be

$$\delta_f = -K\mathbf{x} + \delta_{ff}$$

where $K = [k_1 \quad k_2 \quad k_3 \quad k_4]$ is a feedback gain matrix and δ_{ff} is a feedforward term. Thus,

$$\dot{\mathbf{x}} = (A - B_1 K)\mathbf{x} + B_1\delta_{ff} + B_2\delta_r + B_3\dot{\psi}_{des}$$

Since we don't have knowledge of the value of δ_r, we can only cancel out the desired yaw rate term using the feedforward term. As calculated by Rajamani, the feedforward term is $\delta_{ff} = \dfrac{L}{R} + K_{US}\dfrac{V_x^2}{gR} + k_3 e_{2_{ss}}$, where $e_{2_{ss}} = -\dfrac{\ell_r}{R} + \dfrac{\ell_f}{2C_{\alpha_r}L}\dfrac{mV_x^2}{R}$ is the nominal steady-state yaw angle error. The block diagram of the closed-loop system is shown in Figure 9.

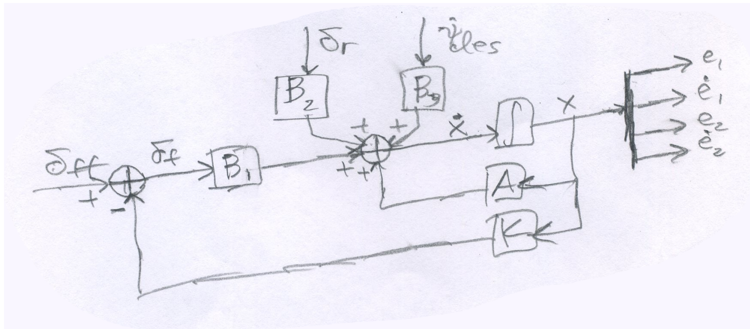

Figure 9: Car Plant and State Feedback Controller

Now we find the steady state error $\mathbf{x}_{ss}$ due to the rear wheels. From Rajamani, we have

$$\mathbf{x}_{ss} = (B_1 K - A)^{-1} \left\{ B_1 \delta_{ss} + B_2 \delta_r + B_3 \frac{V_x}{R} \right\} \tag{8}$$

where $\delta_{ss} = \delta_{ff} - k_3 e_{2_{ss}}$ is the steady state front steering angle. Let $M = (B_1 K - A)^{-1}$ and partition each row such that

$$M = \begin{bmatrix} M_1 \\ \hline M_2 \\ \hline M_3 \\ \hline M_4 \end{bmatrix}$$

We then see that the first element of $\mathbf{x}_{ss}$ is $M_1 \left(B_1 \delta_{ss} + B_3 \frac{V_x}{R} \right) + M_1 B_2 \delta_r$. But $M_1 \left(B_1 \delta_{ss} + B_3 \frac{V_x}{R} \right)$ is 0 since δ_{ss} is defined such that the nominal position error is 0. Therefore, the first element of $\mathbf{x}_{ss}$ is just $M_1 B_2 \delta_r$, which through MATLAB's Symbolic Toolbox is $\frac{k_3 - 1}{k_1} \delta_r$. The other steady-state errors can be found similarly:

$$\mathbf{x}_{ss} = \begin{bmatrix} \dfrac{k_3 - 1}{k_1} \delta_r \\ 0 \\ -\dfrac{\ell_r}{R} + \dfrac{\ell_f}{2 C_{\alpha_r} L} \dfrac{m V_x^2}{R} - \delta_r \\ 0 \end{bmatrix} \tag{9}$$

So to eliminate the steady state position error, we just make $k_3 = 1$. This can be done without knowledge of the path radius R or the rear steering angle δ_r. This means that design of the controller can be done only around convergence speed and fine tuning such that $k_3 = 1$ with the properties of the car. This does however limit the poles that can be placed for the closed-loop system and reduces finding these proper poles to guessing and checking.

Performance of State Feedback Controller

We check the performance of our controller through simulation of our canonical examples.

First, we arbitrarily place the poles at $-1 \pm j$ and $-2 \pm 2j$. Through MATLAB's `place` command, we find that $K = [0.001054 \quad -0.05223 \quad 1.075 \quad -0.1498]$, with k_3 being slightly off from 1. So we won't expect the lateral error to converge to 0.

Simulation results for a straight line input are shown in Figure 10. One marked improvement over the simple compensation scheme is that *both* lateral error and yaw error converge. As can be seen in the car's trajectory, this means that the car will drive straight and parallel to the road. Unfortunately, the lateral errors are on the order of meters, which shows a lack of performance in our first controller design.

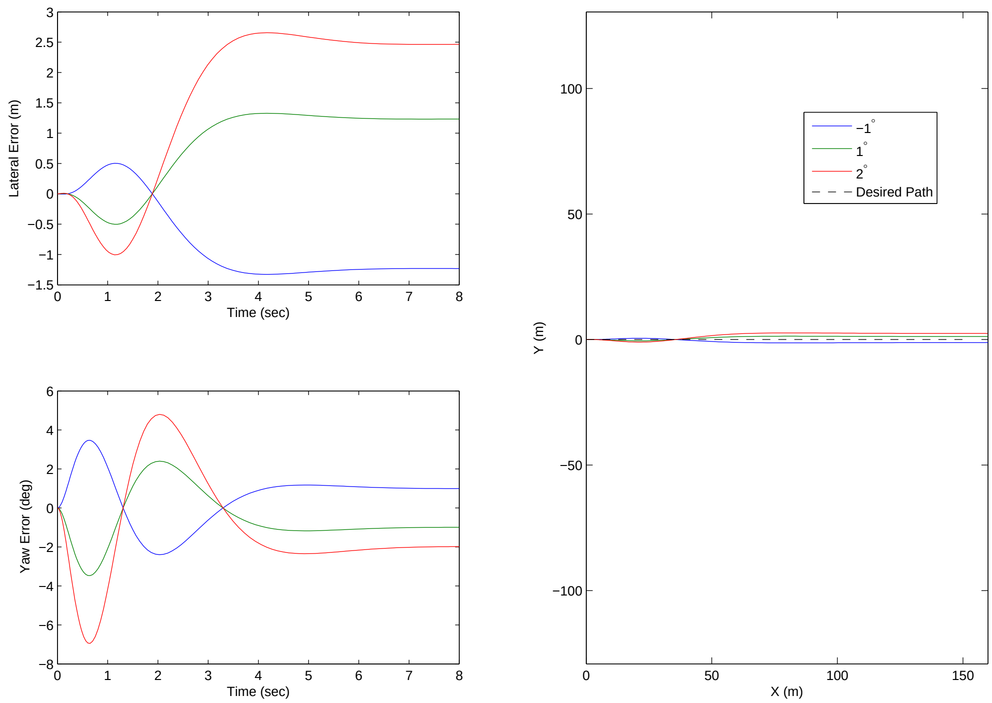

Figure 10: State Feedback Result with Straight Path

We also simulate the results for a circular input, with results being in Figure 11. Steady-state yaw errors match that of Figure 10, whereas steady-state lateral errors are close (in theory they should be the same, it is unknown at this time why they differ). The vehicle again does a much better job of lane keeping but still with a sizable lateral error.

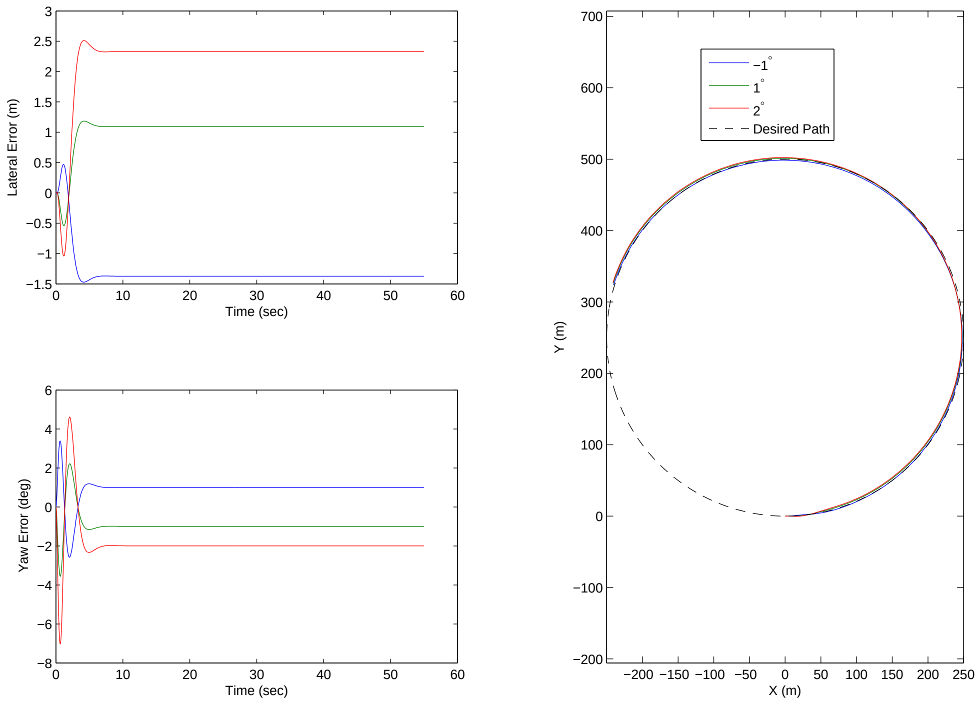

Figure 11: State Feedback Result with Circlular Path

So to attenuate the lateral errors, we will fine tune our controller such that k_3 is very close to 1. When placing the poles at $-1 \pm j$ and $-2.291 \pm 2j$, we find that
$K = [0.0012184 \quad -0.048272 \quad 0.99995 \quad -0.14692]$.

Simulation results for a straight line input are shown in Figure 12. Now, all the lateral errors converge to 0. Unfortunately, in the process, the lateral errors do maximize up to 2 meters before dying away. When actually designing a controller, the peak overshoot must be made as small as possible so that a vehicle doesn't cross into another possibly occupied lane or veer off the road. It is also interesting to note that the yaw errors do not reach zero, and in fact converge to the negative of the rear steering angle. Physically this means that the front and rear steering angles are the same (as discussed in a previous section) and the car body is rotated by the same amount in the opposite direction.

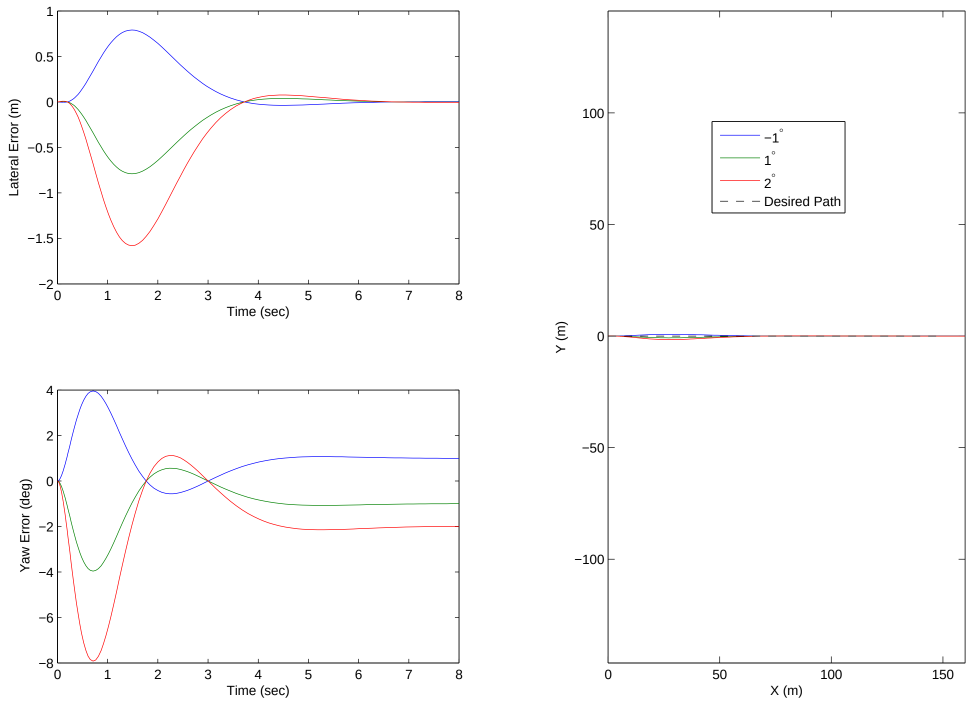

Figure 12: Fine Tuned Feedback Result with Straight Path

Simulation results for a circlular input are shown in Figure 13. The lateral errors all converge to the same small value, but it is nonzero (unknown at this time why). The paths of each vehicle lie very close to the desired path. Just like the previous simulation, there is a significant amount of overshoot. The yaw errors converge to nearly the same values as the previous simulation.

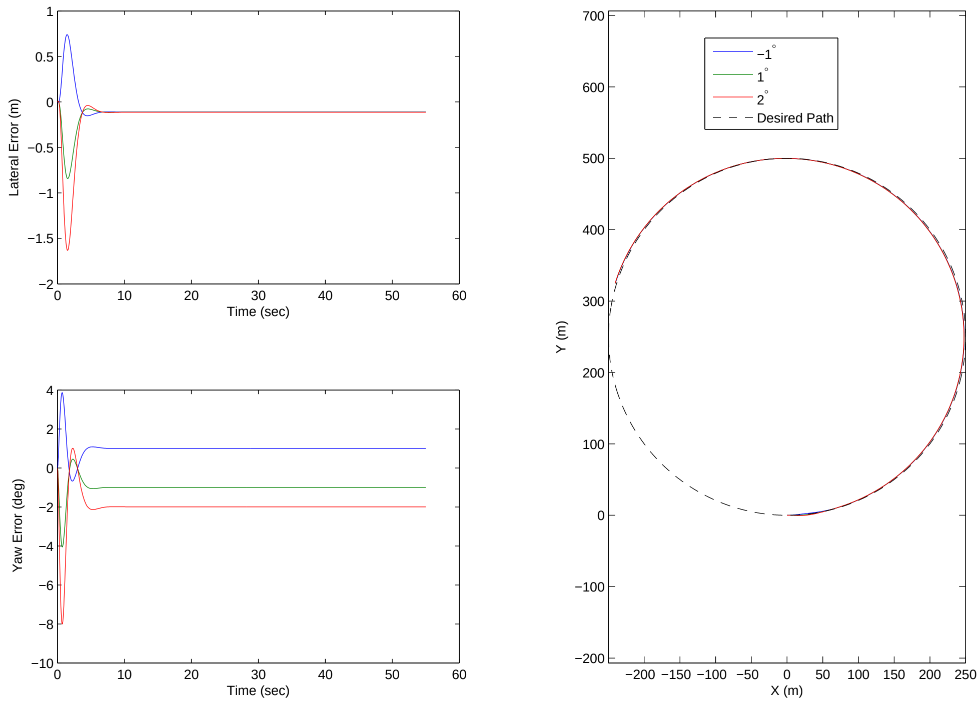

Figure 13: Fine Tuned Feedback Result with Circular Path

Conclusion

In the open loop, a rear steering angle alters the steady-state path of a car, with larger angles causing a more visible change. We found the exact effects that this disturbance has, with simple improvements being proposed (increased speed, offsetting front steering angle). Nonetheless, no open-loop additions improved lane keeping.

So we designed a state feedback controller that would keep the vehicle from diverging. In general, the vehicle *does* converge to some path, but could be significantly off from the desired one. It takes a specially designed controller to have the vehicle drive on (if not very near) to the desired path. The steady state lateral error, nonetheless, is very sensitive to the third controller gain, as evidenced by the lateral errors being on the order of meters when the third gain was 0.075 off from 1.

Our choice of rear steering angle values are most likely outliers but do illustrate the importance of seemingly tiny disturbances. It also showed that one controller design can attenuate all realistic values of this disturbance.

In the future, different controller schemes can be implemented and analyzed. For instance, a look-ahead controller (inclusion of yaw angle error with lateral error) can be designed and compared to our state feedback controller. A controller modeled on a human (lag-lead compensator with a delay) can also be analyzed. Finally, testing our controller with more general paths can give us a better idea of the controller's robustness.

A Deriving Adjusted Radius

We first assume steady state cornering, i.e. $\ddot{\psi} = 0$ and $\ddot{y} = 0$, meaning that the yaw rate and lateral velocities take on constant values $\dot{\psi}_0$ and $\dot{y}_0$, respectively, and that the trajectory of the car will be a circle. Thus, equations 1 and 2 reduce to:

$$a_1\dot{y}_0 + a_2\dot{\psi}_0 = 2(C_{\alpha_f}\delta_f + C_{\alpha_r}\delta_r) \quad (10)$$

$$a_3\dot{y}_0 + a_4\dot{\psi}_0 = 2(\ell_f C_{\alpha_f}\delta_f - \ell_r C_{\alpha_r}\delta_r) \quad (11)$$

where $a_1 = \dfrac{2(C_{\alpha_f} + C_{\alpha_r})}{V_x}$, $a_2 = \dfrac{\frac{1}{2}mV_x^2 + 2(\ell_f C_{\alpha_f} - \ell_r C_{\alpha_r})}{V_x}$, $a_3 = \dfrac{2(\ell_f C_{\alpha_f} - \ell_r C_{\alpha_r})}{V_x}$, and $a_4 = \dfrac{2(\ell_f^2 C_{\alpha_f} + \ell_r^2 C_{\alpha_r})}{V_x}$. To find the steady state radius R, we first note that $\dot{\psi}_0 = \dfrac{V_x}{R}$ and then solve for it from equations 10 and 11:

$$\dot{\psi}_0 = \frac{V_x}{R} = \frac{2a_1(C_{\alpha_f}\delta_f + C_{\alpha_r}\delta_r) - 2a_3(\ell_f C_{\alpha_f}\delta_f - \ell_r C_{\alpha_r}\delta_r)}{a_1a_4 - a_2a_3}$$

Thus,

$$R = \frac{a_1a_4 - a_2a_3}{2a_1(C_{\alpha_f}\delta_f + C_{\alpha_r}\delta_r) - 2a_3(\ell_f C_{\alpha_f}\delta_f - \ell_r C_{\alpha_r}\delta_r)} V_x \quad (12)$$

We then note that

$$\begin{aligned}
a_1a_4 - a_2a_3 &= \frac{4(C_{\alpha_f} + C_{\alpha_r})(\ell_f^2 C_{\alpha_f} + \ell_r^2 C_{\alpha_r}) - 4(\frac{1}{2}mV_x^2 + \ell_f C_{\alpha_f} - \ell_r C_{\alpha_r})(\ell_f C_{\alpha_f} - \ell_r C_{\alpha_r})}{V_x^2} \\
&= \frac{4[\cancel{\ell_f^2 C_{\alpha_f}^2} + \cancel{\ell_r^2 C_{\alpha_r}^2} + (\ell_f^2 + 2\ell_f\ell_r + \ell_r^2)C_{\alpha_f}C_{\alpha_r} - \frac{1}{2}mV_x^2(\ell_f C_{\alpha_f} - \ell_r C_{\alpha_r}) - \cancel{\ell_f^2 C_{\alpha_f}^2} - \cancel{\ell_r^2 C_{\alpha_r}^2}]}{V_x^2} \\
&= \frac{4[(\ell_f + \ell_r)^2 C_{\alpha_f}C_{\alpha_r} - \frac{1}{2}mV_x^2(\ell_f C_{\alpha_f} - \ell_r C_{\alpha_r})]}{V_x^2}
\end{aligned}$$

and that

$$\begin{aligned}
&2a_1(C_{\alpha_f}\delta_f + C_{\alpha_r}\delta_r) - 2a_3(\ell_f C_{\alpha_f}\delta_f - \ell_r C_{\alpha_r}\delta_r) = \\
&\frac{4(C_{\alpha_f} + C_{\alpha_r})(\ell_f C_{\alpha_f}\delta_f - \ell_r C_{\alpha_r}\delta_r) - 4(\ell_f C_{\alpha_f} - \ell_r C_{\alpha_r})(C_{\alpha_f}\delta_r + C_{\alpha_r}\delta_r)}{V_x} = \\
&\frac{4\delta_f(\cancel{\ell_f C_{\alpha_f}^2} + C_{\alpha_f}C_{\alpha_r}\ell_f - \cancel{C_{\alpha_f}^2\ell_f} + C_{\alpha_f}C_{\alpha_r}\ell_r) + \delta_r(-C_{\alpha_f}C_{\alpha_r}\ell_r - \cancel{C_{\alpha_r}^2\ell_r} - C_{\alpha_f}C_{\alpha_r}\ell_f + \cancel{C_{\alpha_r}^2\ell_r})}{V_x} = \\
&\frac{4C_{\alpha_f}C_{\alpha_r}(\ell_f + \ell_r)(\delta_f - \delta_r)}{V_x}
\end{aligned}$$

Note that $L = \ell_f + \ell_r$. Therefore,

$$\begin{aligned}
R &= \frac{C_{\alpha_f}C_{\alpha_r}L^2 - \frac{1}{2}mV_x^2(C_{\alpha_f}\ell_f - C_{\alpha_r}\ell_r)}{C_{\alpha_f}C_{\alpha_r}L(\delta_f - \delta_r)} \\
&= \frac{L}{\delta_f - \delta_r} + \frac{V_x^2}{g(\delta_f - \delta_r)}\left(\frac{mg\ell_r}{2LC_{\alpha_f}} - \frac{mg\ell_f}{2LC_{\alpha_r}}\right)
\end{aligned}$$

Then, noting that $K_{US} = \dfrac{mg\ell_r}{2LC_{\alpha_f}} - \dfrac{mg\ell_f}{2LC_{\alpha_r}}$, we finally get

$$R = \frac{L + K_{US}V_x^2/g}{\delta_f - \delta_r} \quad (13)$$

B Variable Nomenclature as Defined by Rajamani

m	total mass of vehicle
I_{zz}	yaw moment of inertia of vehicle
ℓ_f	longitudinal distance from c.g. to front tires
ℓ_r	longitudinal distance from c.g. to rear tires
L	total wheel base ($\ell_f + \ell_r$)
V_x	longitudinal velocity of c.g. of vehicle
$\dot{y}$	lateral velocity at c.g. of vehicle
ψ	yaw angle of vehicle in global axes
$\dot{\psi}$	yaw rate of vehicle
$\dot{\psi}_{des}$	desired yaw rate from road
C_{α_f}	cornering stiffness of front tire
C_{α_r}	cornering stiffness of rear tire
δ_f	front wheel steering angle
δ_r	rear wheel steering angle
β	slip angle at vehicle c.g.
X, Y	global axes
R	turn radius of vehicle
R_0	turn radius of vehicle assuming $\delta_r = 0$, or radius of road
K_{US}	understeer coefficient
g	acceleration due to gravity
L_e	effective length
δ	effective steady state steering angle
e_1	lateral position error with respect to road
e_2	yaw angle error with respect to road
$\mathbf{x}$	tracking errors on a curve
$\mathbf{x}_{ss}$	steady state tracking errors on a curve
A	system matrix of vehicle
B_1	front steering angle input matrix
B_2	rear steering angle input matrix
B_3	desired yaw rate input matrix
K	feedback matrix
δ_{ff}	feedforward steering angle
δ_{ss}	steady state front steering angle